

REMARKS

Claims 1-30 are pending in the present application. Claims 8 and 23 are cancelled and claims 1, 3, 4, 9, 14, 15, 16, 18, 19, 24, 29 and 30 are amended. Claims 31 and 32 are added. Independent claims 1, 14, 16 and 29 are amended to include the content of cancelled dependent claims 8 and 23. Claims 3, 4, 9, 14, 15, 16, 18, 19, 24 and 30 are amended to provide antecedent basis and to correct typographical errors. Claim 16 is amended for clarity and consistency with claim 1 by replacing both occurrences of the word "converting" in line 11 with the word "serializing". Support for new claims 31 and 32 may be found at least on page 2, line 32 through page 3, line 4; on page 12, line 15 through page 13, line 3; on page 14 lines 19-22; and on page 20, lines 17-23.

Reconsideration of the claims is respectfully requested.

Also, Applicants will submit, under separate cover, another set of formal drawings as requested in the Notice of Draftsperson's Patent Drawing Review, which states that the bottom parts were cut off in scanning.

Applicants also wish to thank Examiner Li B. Zhen for granting a telephone interview for Tuesday, March 22, 2005 to discuss the following remarks.

I. Objection to Claims

The Office Action objects to claims 4 and 19 because of the following informalities: "converting step ad the binding step" [lines 1-2]. In response, claims 4 and 19 are amended to correct the typing error by replacing "ad" with "and". Applicants respectfully submit that the amendment to claims 4 and 19 overcomes this objection.

II. 35 U.S.C. § 112, Second Paragraph

The Office Action rejects claims 3 and 18 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter, which Applicants regard as the invention. This rejection is respectfully traversed. The Office Action states:

Claims 3 and 18 recite the limitation "the locating step" and "the locating means" in lines 1-2. There is insufficient antecedent basis for this limitation in the claim.

Office Action dated December 14, 2004, page 2.

Claims 3 and 18 are amended by replacing the phrase "the locating step" with "the searching step" and by replacing the phrase "the locating means" with "the searching means" in claims 3 and 18, respectively. Therefore, Applicants respectfully submit that the rejection of claims 3 and 18 under 35 U.S.C. § 112, second paragraph has been overcome.

III. 35 U.S.C. § 101

The Office Action rejects claims 1-13 under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. This rejection is respectfully traversed.

The Office Action states:

Claims 1-13 are directed to method steps which can be practiced mentally in conjunction with pen and paper, therefore they are directed to non-statutory subject matter. Specifically, as claimed, it is uncertain what performs each of the claimed method steps. Moreover, each of the claimed steps, inter alia, collecting, forwarding, searching, attaching, redirecting, converting, binding, receiving, converting and serializing, can be practiced mentally in conjunction with pen and paper. The claimed steps do not define a machine or computer implemented process [see MPEP2106]. Therefore, the claimed invention is directed to non-statutory subject matter. (The examiner suggests applicant to change "method" to "computer implemented method" in the preamble to overcome the outstanding 35 U.S.C. 101 rejection).

Office Action dated December 14, 2004, page 3.

Applicants respectfully submit that the claims recite a "method in a data processing system" and therefore, are not practice mentally in conjunction with pen and paper. Further, Figure 1 and Figure 2 illustrate a data processing system implemented as a server and a client, respectively, in which the present invention may be implemented. Therefore, Applicants respectfully submit that the rejection of claims 1-13 under 35 U.S.C. § 101 has been overcome.

IV. 35 U.S.C. § 102, Alleged Anticipation Based on *Quirt*

The Office Action rejects claims 9, 11-13, 15, 24, 26-28 and 30 under 35 U.S.C. § 102(e) as being allegedly anticipated by *Quirt*, U.S. Patent Number 6,269,378. This rejection is respectfully traversed.

As to independent claims 9, 15, 24, and 30, the Office Action states:

As to claim 9, Quirt teaches a method in a data processing system for obtaining object references, the method comprising:

receiving a request for an object reference [When the Local Name Service 402 receives the message from the software object A 400; col. 10, lines 23-57], wherein the request includes a source name space path, and identification of a destination, and a destination name space path [When the Local Name Service 402 receives the message from the software object A 400 containing the fake reference, the Local Name Service checks whether the off-node Name Service 404 has responded to the off-node request; col. 10, lines 23-57];

searching a name space for the object reference using the source name space path [off-node Name Service 404 performs its search operation 416; col. 10, lines 29-30]; and

responsive to locating the object reference [off-node Name Service 404 sends back to the Local Name Service 402 message 418 with the reference to the software object B 406; col. 10, lines 31-33], sending the object reference to a destination using the locate message 414 the same fake object reference that it issued in the acknowledgement message 412 sent to the software object A 400. The Local Name Service 402 updates the corresponding record in its data structure 160; col. 10, lines 33-57], wherein the destination uses the destination name space path to bind the object reference [The Local Name Service 402 is now able to provide an immediate valid response to a future query for look-up of the formerly missing persistent name of B 406; col. 10 lines 35-50]. ...

As to claim 15, this is a system claim that corresponds to method claim 9; note the rejection to claim 9 above, which also meets this system claim.

As to claims 24 and 26-28, these are system claims that correspond to method claims 9 and 11-13; note the rejection to claims 9 and 11-13 above, which also meet these system claims.

As to claim 30, this is a product claim that corresponds to method claim 9; note the rejection to claim 9 above, which also meets this product claim.

Office Action dated December 14, 2004, pages 4-5.

Claim 9, which is representative of the other rejected independent claims 15, 24 and 30 with regard to similarly recited subject matter, reads as follows:

9. A method in a data processing system for obtaining object references, the method comprising:

receiving a request for an object reference, wherein the request includes a source name space path, an identification of a destination, and a destination name space path;

searching a name space for the object reference using the source name space path; and

responsive to locating the object reference, sending the object reference to a destination using the identification of the destination, wherein the destination uses the destination name space path to bind the object reference. (emphasis added)

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). Applicants respectfully submit that *Quirt* does not identically show every element of the claimed invention arranged as they are in the claims. Specifically, *Quirt* does not teach or suggest receiving a request for an object reference, wherein the request includes a source name space path, an identification of a destination, and a destination name space path, as recited in independent claims 9, 15, 24 and 30. Additionally, *Quirt* does not teach or suggest that responsive to locating the object reference, sending the object reference to a destination using the identification of the destination, wherein the destination uses the destination name space path to bind the object reference, as recited in independent claims 9, 15, 24 and 30.

Quirt is directed toward a Name Service for providing client software objects with the software object reference of other software objects in a software system. Look-up requests to the Name Service are made by the objects as ordinary synchronous procedure calls. If a client requests the software object reference of a software object from the Name Service and that request cannot be satisfied by the tables of the Local Name Service, a fake software object reference is returned to the client. This allows look-up requests to terminate within a given time frame. The fake software object reference actually points back to the Name Service. When the client attempts to send a message to the software object, it does so in an asynchronous fashion. The object saves its current context and relinquishes control of the process thread and waits for a reply to its message. The Local Name Service receives the message and sends an off-node query to attempt to locate the software object reference. When it receives a reply to the query, it returns an error message to the client causing the client to repeat the look-up process. The Local Name Service is now ready to respond to the query and returns to the client the software object reference of

the software object sought. *Quirt* does not teach or suggest receiving a request for an object reference, wherein the request includes a source name space path, an identification of a destination, and a destination name space path, as recited in claims 9, 15, 24 and 30. To the contrary, *Quirt* teaches that the chain of Name Services tries to resolve the object reference. *Quirt* teaches breaking the transaction with a fake object reference to release the shared execution tread so that chained Name Services can find the actual software object reference. Thus, there is no motivation in *Quirt* to provide a source name space path or a destination name space path. Further, *Quirt* does not teach or suggest that responsive to locating the object reference, sending the object reference to a destination using the identification of the destination, wherein the destination uses the destination name space path to bind the object reference, as recited in claims 9, 15, 24 and 30. To the contrary, *Quirt* teaches returning a fake object reference to a local name service.

The Office Action refers to the following portion of *Quirt* in the rejection of independent claims 9, 15, 24 and 30:

When the Local Name Service 402 receives the message from the software object A 400 containing the fake reference, the Local Name Service checks whether the off-node Name Service 404 has responded to the off-node request. If no response has been received, the Local Name Service saves the message 520 as information associated with the appropriate entry in data structure 160. The off-node Name Service 404 performs its search operation 416. At some point, the off-node Name Service 404 sends back to the Local Name Service 402 message 418 with the reference to the software object B 406. In a specific example, in order to properly associate the response 418 with message 414, and with the corresponding entry in data structure 160, the local Name Service 402 uses as its own identity in the locate message 414 the same fake object reference that it issued in the acknowledgement message 412 sent to the software object A 400. The Local Name Service 402 updates the corresponding record in its data structure 160. It may optionally also update its primary data structure 150 to contain this same persistent name and object reference for object B 406. The Local Name Service 402 is now able to provide an immediate valid response to a future query for look-up of the formerly missing persistent name of B 406. It is also now able to complete its handling of the message 520. To do this, the Local Name Service 402 sends an error message 422 to object A 400, with an error code indicating that the object reference used in message 420 is an obsolete or invalid object reference. The message 422 is received in the regular message queue of object A. Object A is given control of the process thread and receives the message 422. The error message 422 causes Software object A 400 to restart the procedure of invoking the Local Name Service 402 to effect the object reference resolution.

In other words, the software object A 400 will build and send a new locate message 424.

Quirt, column 10, lines 23-57.

This portion of *Quirt* teaches that an off-node Name Service, from the chain of Name Services, performs a search for an object reference that was not in the Local Name Service. At some point, the off-node Name Service locates and returns the object reference to the Local Name Service. As discussed above, *Quirt* does not teach or suggest receiving a request for an object reference, wherein the request includes a source name space path, an identification of a destination, and a destination name space path, as recited in claims 9, 15, 24 and 30. Additionally, *Quirt* does not teach or suggest that responsive to locating the object reference, sending the object reference to a destination using the identification of the destination, wherein the destination uses the destination name space path to bind the object reference, as recited in claims 9, 15, 24 and 30. In claims 9, 15, 24 and 30, the object reference is located in the source name space path and sent to the destination using the identification of the destination in the original request. The mechanism of the present invention bypasses the problematic use of non-standard name space bootstrap protocols by Object Request Brokers for obtaining remote object references.

In view of the above, Applicants respectfully submit that *Quirt* does not teach each and every feature of independent claims 9, 15, 24 and 30, as is required under 35 U.S.C § 102(e). *Quirt* does not teach each and every feature of dependent claims 11-13 and 26-28 at least by virtue of their dependency on claims 9, 15, 24 and 30, respectively. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 9, 11-13, 15, 24, 26-28 and 30 under 35 U.S.C § 102(e).

In addition to the above, *Quirt* does not teach the specific features recited in dependent claims 11-13 and 26-28. With respect to claims 11 and 26, *Quirt* does not teach or suggest that the identification of the destination is a universal resource locator. As stated previously, *Quirt* does not teach including an identification of the destination in the request for an object reference and therefore, does not teach that the identification is a URL. In addition, with respect to claims 12 and 27, *Quirt* does not explicitly teach or suggest that the request is a POST request.

Additionally, with respect to claims 13 and 28, *Quirt* does not teach or suggest converting the object reference to a standard common object request broker architecture object prior to sending the object reference to the destination. In the rejection of claims 13 and 18, the Office Action refers to the following portion of *Quirt*:

In a preferred embodiment, as shown in FIG. 7, the Name Service is implemented using a suitable programming language. In a specific example, the Name Service is implemented using C++ with CORBA interfaces, and is run on a Unix or Vxworks platform. The Name Service in accordance with the spirit of the invention may run on any suitable computing platform that constitutes a node of the network. As shown in FIG. 7 such computing platform typically include a processor 902 in operative relationship with a memory 904 containing program instructions implementing the module described in this specification. The computer program 904 may be stored on any suitable computer readable medium 900.

Quirt, column 11, lines 19-31.

This portion of *Quirt* only teaches that the Name Service may be implemented using CORBA. *Quirt* does not mention converting the object reference to a standard common object request broker architecture object prior to sending the object reference to the destination, as recited in claims 13 and 18. The method of the present invention supports vendors that do not use the same protocol for obtaining a reference to a name service and that may use different Object Request Brokers.

Thus, in addition to being dependent on their respective independent claims 9, 15, 24 and 30, dependent claims 11-13 and 26-28 are also distinguished over the *Quirt* reference based on the specific features recited therein.

V. **35 U.S.C. § 103, Alleged Obviousness Based on *Quirt* and *Kukura***

The Office Action rejects claims 1-8, 10, 14, 16-23, 25 and 29 under 35 U.S.C. § 103(a) as being allegedly unpatentable over *Quirt* in view of *Kukura et al.* (U.S. Patent 6,633,923), hereinafter referred to as *Kukura*. This rejection is respectfully traversed.

As to independent claims 1, 14, 16 and 29, the Office Action states:

As to claim 1, *Quirt* teaches the invention substantially as claimed including a method in a data processing system for binding object references from a remote name space into a local name space, the method comprising:

collecting information to create a request to bind an object reference [software object in the software system registers with the Name Service by sending

its persistent Name, software object reference and other relevant information to the Name Service in the form of a registration message; col. 8, lines 22-33];

forwarding the request to a source application server [software object 300 sends a registration message 308 to the Name Service requesting that a registration be made. The registration message 308 includes a scope parameter indicating that the registration is to be performed at a central level (in the Central Name Service, in the Local Name Service 302 and in the associated Cluster Name Service); col. 8, lines 40-48];

searching for the object reference in the remote name space [Name Service can perform remote searching to translate the persistent name to a suitable software object reference; col. 5, lines 10-18];

responsive to locating the object reference attaching the interoperable object reference to the request [Local Name Service 302 then sends a message 312 its associated Cluster Name Service 304; col. 8, lines 49-62];

redirecting the request to a destination application server [Local Name Service 302 then sends a message 312 its associated Cluster Name Service 304 requesting that the software object reference be registered; col. 8, lines 49-62]; and

binding the object reference into the local name space on the destination application server [Cluster Name Service 304 creates the same entry 314 in its table; col. 8, lines 49-62].

Although Quirt teaches the invention substantially as claimed, Quirt does not specifically teach serializing an object reference and converting a serialized interoperable object reference back to an object reference.

However, Kukura teaches serializing an object reference and converting a serialized interoperable object reference back to an object reference [Optimize the performance of marshaling and demarshaling (and therefore conversion to and from strings) of IORs; col. 43, lines 45-48].

It would have been obvious to a person of ordinary skill in the art at the time of the invention to apply the teachings of serializing an object reference and converting a serialized interoperable object reference back to an object reference as taught Kukura to the invention of Quirt because this formats the request message in such a way that all the interconnected computers can understand and respond to the request message [col. 1, lines 29-45 of Kukura]. ...

As to claim 14, this is a system claim that corresponds to method claim 1; note the rejection to claim 1 above, which also meets this system claim. Examiner notes that a bus system, a communications unit connected to the bus system, and a memory connected to the bus system are inherent to a computer system.

As to claims 16-23, these are system claims that correspond to method claims 1-8; note the rejection to claims 1-8 above, which also meet these system claims. ...

As to claim 29, this is a product claim that corresponds to method 1; note the rejection to claim 1 above, which also meets this product claim.

Office Action dated December 14, 2004, pages 6-10.

As amended, claim 1, which is representative of the other rejected independent claims 14, 16 and 29 with regard to similarly recited subject matter, reads as follows:

1. A method in a data processing system for binding object references from a remote name space into a local name space, the method comprising:
 - collecting information to create a request to bind an object reference,
wherein the request includes an identification of a source, a source name space path, an identification of a destination, and a destination name space path used to bind the object reference;
 - forwarding the request to a source application server;
 - searching for the object reference in the remote name space;
 - responsive to locating the object reference, serializing the object reference to a serialized interoperable object reference;
 - attaching the serialized interoperable object reference to the request;
 - redirecting the request to a destination application server;
 - converting the serialized interoperable object reference back to the object reference; and
 - binding the object reference into the local name space on the destination application server. (emphasis added)

Neither *Quirt* nor *Kukura*, taken individually or in combination, teaches or suggests collecting information to create a request to bind an object reference, wherein the request includes an identification of a source, a source name space path, an identification of a destination, and a destination name space path used to bind the object reference, as recited in claims 1, 14, 16 and 29.

As discussed above, *Quirt* does not teach or suggest collecting information to create a request to bind an object reference, wherein the request includes an identification of a source, a source name space path, an identification of a destination, and a destination name space path used to bind the object reference. Further, *Kukura* does not provide for the deficiencies of *Quirt*.

Kukura is directed to a method and system for dynamic configuration of interceptors in a client-server environment. The method comprises the steps of intrinsically chaining the interceptors, and storing state information, in at least one of the chained interceptors, directed to a reference to the next interceptor. The interceptors provide hooks to programmers to execute their piece of code at certain points during Object Request Brokering. Typical uses of the interceptors include: transaction service integration, security message compression and encryption, fault tolerance and other operations such as tracing, profiling, debugging, and logging. *Kukura* is cited for

teaching marshaling and demarshing of IORs. *Kukura* does not teach or suggest collecting information to create a request to bind an object reference, wherein the request includes an identification of a source, a source name space path, an identification of a destination, and a destination name space path used to bind the object reference, as recited in claims 1, 14, 16 and 29.

Quirt and *Kukura* do not teach or suggest collecting information to create a request to bind an object reference, wherein the request includes an identification of a source, a source name space path, an identification of a destination, and a destination name space path used to bind the object reference. Therefore, the alleged combination of *Quirt* and *Kukura* does not teach or suggest this feature, as recited in independent claims 1, 14, 16 and 29.

Further, *Quirt* and *Kukura*, either taken alone or in combination, do not teach or suggest the features of dependent claims 2-8, 10, 17-23 and 25 at least by virtue of their dependency on claims 1, 14, 16 and 29, respectively. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 1-8, 10, 14, 16-23, 25 and 29 under 35 U.S.C. § 103(a).

VI. New Claims 31 and 32

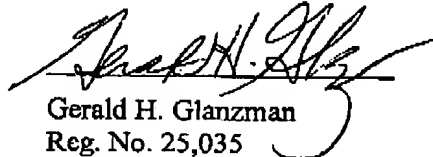
In addition to the above, the cited references do not teach the specific features recited in new dependent claims 31 and 32. Specifically, *Quirt*, *Kukura*, and the alleged combination of these references do not teach or suggest that the local name space uses a different object request brokering architecture than the remote name space, as recited in claim 31, and that the name space of the source name space path uses a different object request brokering architecture than a destination name space of the destination name space path, as recited in claim 32. Thus, in addition to being dependent upon independent claims 1 and 9, respectively, claims 31 and 32 are also distinguished over the cited references based on the specific features recited therein.

VII. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: March 14, 2005

Respectfully submitted,



Gerald H. Glanzman
Reg. No. 25,035
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants

GHG/vja